

In paragraaf 3.2.3 is een aantal verbeteringen aangebracht om de tekst en het stuf0302.xsd met elkaar in overeenstemming te brengen.

De voorbeelden zijn in overeenstemming gebracht met de GAB definitie van onvolledige datum en tijd.

- [RFC0461](#), kerngegevens verplicht ook al is sleutelOntvangend gevuld

De specificaties dat de kerngegevens in een kennisgeving niet verplicht zijn als sleutelOntvangend is gevuld verwijderd, evenals de uitzonderingen op deze regel.

NB: In het kader van de versimpeling van de standaard zijn in paragraaf 6.4.1 en 64.2 de voorwaarden voor het opnemen van een object c.q. een relatie in een antwoordbericht verwijderd.

- [RFC0447](#), Vrije structuur in synchrone vrije berichten

In het begin van paragraaf 7.2 zijn de voorwaarden gedefinieerd voor het niet gebruiken van de door StUF gedefinieerde structuren en functionaliteit in vrije berichten.

- [RFC0417](#), Sorteringen op meervoudig voorkomende elementen of relaties

In paragraaf 6.4.1 is voor de laatste alinea tekst toegevoegd die specificeert hoe wordt omgegaan met sorteringen op meervoudig voorkomende elementen of relaties

- [RFC0413](#), Overbodig maken nillable="true", [RFC0121](#); Gelijktrekken datum/tijd definities met GAB en [RFC0478](#) Tijdzone in tijdstipBericht?

De constructie met het element leeg binnen relaties verwijderd en vervangen door het niet langer verplicht zijn van de sequence op het hoogste niveau binnen een relatie. Dit heeft consequenties voor paragraaf 4.3.2 en de voorbeelden.

Binnen de onvolledige datum/tijd is de choice op het hoogste niveau optioneel gemaakt, zodat nillable="true" niet langer nodig is voor een onvolledige datum/tijd. Daarnaast de gewone datum/tijd restricted van onvolledige datum/tijd. Dit heeft consequenties voor de voorbeelden.

Bovenaan in paragraaf 3.2.3 is aangegeven hoe in een StUF-bericht een datum/tijd zonder tijdzone geïnterpreteerd moet worden.

- RFC0473: Hernoemen kerngegevens naar matchgegevens

Overall waar in de tekst kerngegeven(s) staat is dit vervangen door matchgegeven(s).

In het schema is in de enumeration voor het simpleType StUFScope allesMaarKerngegevensGerelateerden vervangen door allesMaarMatchgegevensGerelateerden en kerngegevens door matchgegevens.

In deze versie is ook de aanvullendeElementen definitie overgenomen van StUF0301 en zijn binnen de paragrafen 3.2.4.1 (extraElementen) en 3.2.4.2 (aanvullendeElementen) de StUF namespaces aangepast naar StUF0302.

- RFC0437: De inspringing in de lijst op p. 127 verbeterd.

- RFC0345: Patchnummer

Er is nog slechts één attribute patch met het patchnummer van het koppelvlak toegestaan. Het attribute patch is verwijderd in stuf0302.xsd en vervangen door het simpleType Patchnummer.

- Verbeteren foutjes in definitie synchronisatieberichten

Er zijn nog enkele fouten in de definitie van synchronisatieberichten verbeterd. Belangrijkste aanpassing is het verwijderen van het attribute functie binnen de elementen <toevoeging>, <wijziging> en de elementen voor het toevoegen van gerelateerden.

- Herschrijven hoofdstuk 7 Vrije berichten

Dit hoofdstuk is uitgebreid met een aantal voorbeelden en tekstueel verbeterd. Er is ook een belangrijke inhoudelijke wijziging doorgevoerd. Het gebruik van StUF:aanvullendeElementen is weer teruggedraaid. Extra gegevens worden nu weer gespecificeerd in gereserveerde elementen. De reden hiervoor is dat in een vrij bericht design time de functionaliteit bekend is en dus expliciet gespecificeerd kan worden in plaats van impliciet via de StUF:aanvullendeElementen constructie.

- Het attribute verwerkingssoort is verwijderd in de vraag-om-synchronisatie berichten.

- In hoofdstuk 7 de syntax voor een entiteit-element aangepast.

- In hoofdstuk 5 fouten verbeterd in de beschrijving structuur Sh01/2-bericht.

- Verdwenen sequence diagrammen zijn hersteld.

- In paragraaf 4.1.1 is nog een kleine correctie aangebracht n.a.v. het al eerder verwijderen van het attribute 'patch' in het stuf0302 schema.

- [RFC0498](#): In hoofdstuk 7 zijn wijzigingen aangebracht m.b.t. de wens het attribute 'entiteittype' niet meer te definiëren op elementen met een vrij te kiezen naam en een 'functie' attribute.

- [RFC0496](#): [Sta opnemen vrij bericht in een ander vrij bericht toe](#)
[In hoofdstuk 7 is de uitwerking van dit RFC toegevoegd en in stuf0302.xsd is 'dienstvrij' toegevoegd aan de enumeratie](#)

De standaard eist dat de gebruikelijke StUF-berichtstructuur doorloopt binnen StUF:aanvullendeElementen. Voor elementen binnen stuf:aanvullendeElementen/ae:bankgegevens gelden dezelfde regels als de andere elementen van de StUF-entiteit waarbinnen het element <StUF:aanvullendeElementen> voorkomt. Het komt erop neer dat de tags <StUF:aanvullendeElementen>, <ae:bankgegevens>, </ae:bankgegevens>, <ae:mijnLocatie>, </ae:mijnLocatie> en </StUF:aanvullendeElementen> weggedacht kunnen worden en dat er dan een gewone StUF-entiteit overblijft met alle regels die daarvoor gelden.

Hieronder een voorbeeld van een schema waarmee de extra elementen in bovenstaand voorbeeld gevalideerd worden:

```
<schema xmlns="http://www.w3.org/2001/XMLSchema"
xmlns:ae="http://www.egem.nl/StUF/sector/bg/0310/aanvullendeElementen"
xmlns:StUF="http://www.stufstandaarden.nl/onderlaag/stuf0302"
xmlns:GML="http://www.opengis.net/gml"
targetNamespace="http://www.egem.nl/StUF/sector/bg/0310/aanvullendeElementen"
elementFormDefault="qualified" attributeFormDefault="unqualified" version="031003">
  <import namespace="http://www.stufstandaarden.nl/onderlaag/stuf0302"
  schemaLocation="stuf0302.xsd"/>
  <import namespace="http://www.opengis.net/gml" schemaLocation="geometryBasic0d1d.xsd"/>
  <element name="aanvullendeElementen" type="ae:AanvullendeElementen"/>
  <complexType name="AanvullendeElementen">
    <sequence>
      <element name="bic" type="ae:Bic-e" nillable="true" minOccurs="0"/>
      <element name="iban" type="ae:Iban-e" nillable="true" minOccurs="0"/>
    </sequence>
  </complexType>
  <complexType name="Bic-e">
    <simpleContent>
      <extension base="ae:Bic">
        <attributeGroup ref="StUF:element"/>
      </extension>
    </simpleContent>
  </complexType>
  <complexType name="Iban-e">
    <simpleContent>
      <extension base="ae:Iban">
        <attributeGroup ref="StUF:element"/>
      </extension>
    </simpleContent>
  </complexType>
  <simpleType name="Bic">
    <restriction base="string">
      <maxLength value="11"/>
    </restriction>
  </simpleType>
  <simpleType name="Iban">
    <restriction base="string">
      <maxLength value="34"/>
    </restriction>
  </simpleType>
</schema>
```

3.2.3 Datum en tijd

Datum en tijd worden in StUF0302 opgenomen zoals voorgeschreven vanuit de Gemeenschappelijk Afspraken Basisregistraties (GAB). Voor een datum/tijd definieert de StUF-standaard een restriction op `xs:dateTime` die het tijdstip 24:00:00 verbiedt en die de nauwkeurigheid van een tijdstip beperkt tot één milliseconde, doordat een tijdstip maar drie cijfers na de decimale punt mag bevatten. Het gebruik van de tijdzone is optioneel. Als de tijdzone ontbreekt, dan is de tijdzone de tijdzone voor Nederland, zoals die op dat tijdstip gold (als de zomertijd geldt dus UTC+2 en als de wintertijd geldt dus UTC+1). Voor het uur overlap bij het terugzetten van de klok van zomertijd naar wintertijd is de tijd UTC+2. Het tijdvak 03:00 - 04:00 UTC+2 kan hierdoor alleen gespecificeerd worden door het expliciet opnemen van een tijdzone. Deze keuze is gemaakt, omdat veel systemen die StUF gebruiken nooit een tijdzone hebben geregistreerd. Deze keuze waarborgt daarnaast de volgordelijkheid van de berichten.

Een element voor een datum of datum/tijd waarin de maand of de dag binnen een maand onbekend kan zijn, wordt opgenomen conform de voorschriften in het document

http://www.noraonline.nl/images/noraonline/c/c3/GAB_mogelijk_onvolledige_datum_1.0.pdf. Om deze voorschriften te implementeren zijn binnen de StUF0302 namespace de complexTypes DatumMogelijkOnvolledig-e, DatumMogelijkOnvolledig-r,

TijdstipMogelijkOnvolledig-e en TijdstipMogelijkOnvolledig-r gedefinieerd en gebruikt binnen de elementdefinities voor tijdvakGeldigheid, ~~tijdvakObject~~ en tijdvakRelatie.

Omdat de complexTypes DatumMogelijkOnvolledig-e of TijdstipMogelijkOnvolledig-e het attribute noValue bevatten, kan het onbekend zijn van het jaar gespecificeerd worden door een element met één van deze complexTypes op te nemen met noValue="waardeBestaat". De complexTypes DatumMogelijkOnvolledig-e of TijdstipMogelijkOnvolledig-e mogen alleen gebruikt worden als de datum of datum/tijd daadwerkelijk onvolledig mag zijn.

Een element waarin de datum of datum/tijd geen onvolledige waarde bevatten, wordt altijd opgenomen met als complexType Datum-e, Datum-r respectievelijk Tijdstip-e en Tijdstip-r. Deze complexTypes zijn restrictions van TijdstipMogelijkOnvolledig-e.

3.2.4 Dynamische waardenlijsten

Het gebruik van een enumeratie gedefinieerd in een xsd-bestand heeft als nadeel dat wijziging alleen mogelijk is door over te gaan naar een nieuwe versie van het sectormodel. Dit is wel erg zwaar voor een simpele wijziging in een enumeratie. Zodra er een kans is dat een enumeratie zal wijzigen gedurende de verwachte levensduur van het sectormodel, is een ander mechanisme gewenst.

De StUF-standaard biedt dit mechanisme in de vorm van het aan de IMGEO- en gml-standaarden ontleende attribute codeSpace met als type het simpleType anyURI gedefinieerd in XML Schema. De uri in codeSpace dient te verwijzen naar de lijst met mogelijke waarden voor het element waarop het attribute codeSpace is gedefinieerd. StUF definieert het attribute codeSpace niet in haar eigen namespace, omdat het dan niet mogelijk is om via een enumeratie een vaste waarde voor het attribute te definiëren. De ontwerper van een sectormodel kan het attribute codeSpace opnemen op elementen met een dynamische waardenlijst. Als de ontwerper verwacht dat de url van de waardenlijst gedurende de levensduur van het sectormodel hetzelfde blijft, dan is het verstandig om die url in codeSpace als enig mogelijke waarde te definiëren en om codeSpace verplicht te maken.

3.3 Metagegevens

Een object heeft eigenschappen en de waarden van die eigenschappen worden als gegevens van dat object vastgelegd. Bij een persoon zijn de geslachtsnaam (Broek), het e-mailadres (broek009@onsnet.nu) en de geboortedatum (12-10-1958) voorbeelden van gegevens. Ook gegevens of groepen van gegevens kunnen eigenschappen hebben, bijvoorbeeld de periode waarin een bepaalde waarde geldig is (geweest), het feit dat een bepaald gegeven in onderzoek is (geweest) of het brondocument waaraan de gegevens zijn ontleend. Dit soort gegevens worden ook wel metagegevens genoemd. Vanuit het informatiemodel is bekend welke elementen in een entiteit metagegeven zijn.

Metagegevens zijn net zoals de sleutels onafhankelijk van een concreet entiteitstype als Verblifsobject of Persoon en kunnen dus los van een concreet sectormodel gedefinieerd worden. De StUF-standaard doet dit en definieert voor een relatief brede verzameling metagegevens de betekenis en de functionaliteit. Op deze wijze is er een uniform mechanisme voor het in berichten opnemen van metagegevens. Bij het definiëren van metagegevens heeft de StUF-standaard als uitgangspunt genomen de metagegevens onderkend in de GBA en in de BAG. De StUF-standaard definieert wel van de GBA en BAG afwijkende mechanismen, omdat in StUF een zo eenvoudig en natuurlijk mogelijk gebruik van metagegevens in berichten voorop staat. Twee ontwerpcriteria liggen aan de gemaakte keuzen ten grondslag:

1. Metagegevens zijn een uitbreiding op een entiteitstype. Het toevoegen van metagegevens aan een entiteitstype mag geen consequenties hebben voor de structuur van dat entiteitstype in een bericht anders dan het toevoegen van elementen en attributes voor de metagegevens.
2. Als er voor een entiteitstype metagegevens zijn gedefinieerd, dan dienen gebruikers van berichten voor een entiteitstype met metagegevens zonder problemen ook berichten zonder metagegevens te kunnen maken c.q. dienen ze bij de verwerking de metagegevens eenvoudig te kunnen negeren. StUF heeft de ambitie om met één set objectdefinities zowel simpele als complexe functionaliteit te kunnen ondersteunen, bijvoorbeeld simpele webservices voor het opvragen van een klein setje actuele gegevens en complexe webservices voor het opvragen van een persoon inclusief zijn materiële en formele historie en inclusief brondocumenten.

De StUF-standaard onderkent de volgende groepen metagegevens: